

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

NOTICE OF APPEAL FROM THE EXAMINER
TO THE BOARD OF APPEALS

Applicant(s): Soria et al.
Serial No.: 09/960,769
For: IMPLEMENTING VERSIONING SUPPORT FOR DATA USING A TWO-
TABLE APPROACH THAT MAXIMIZES DATABASE EFFICIENCY
Filed: September 21, 2001
Examiner: C. B. Paula
Art Unit: 2178
Confirmation No.: 6311
Customer No.: 27623 Attorney Docket No.: STL920000113US1

Mail Stop Appeal Brief-Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

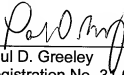
We are enclosing for filing in the above-identified application the following:

1. Appellant's Appeal Brief (*in triplicate*);
2. Transmittal letter in duplicate; and
3. Postcard.

Please charge the fee of \$510.00 and any additional fees or credit any such fees, if necessary to Deposit Account No. **09-0460** in the name of International Business Machines Corporation. A duplicate copy of this sheet is attached.

Respectfully submitted,

November 30, 2007
Date



Paul D. Greeley
Registration No. 31,019
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.
One Landmark Square, 10th Floor
Stamford, CT 06901-2682
Telephone: (203) 327-4500
Telefax: (203) 327-6401

NOTICE OF APPEAL FROM THE EXAMINER
TO THE BOARD OF APPEALS

Applicant(s): Soria et al.
Serial No.: 09/960,769
For: IMPLEMENTING VERSIONING SUPPORT FOR DATA USING A TWO-
TABLE APPROACH THAT MAXIMIZES DATABASE EFFICIENCY
Filed: September 21, 2001
Examiner: C. B. Paula
Art Unit: 2178
Confirmation No.: 6311
Customer No.: 27623 Attorney Docket No.: STL920000113US1

Mail Stop Appeal Brief-Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

We are enclosing for filing in the above-identified application the following:

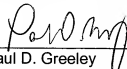
1. Appellant's Appeal Brief (*in triplicate*);
2. Transmittal letter in duplicate; and
3. Postcard.

Please charge the fee of \$510.00 and any additional fees or credit any such fees, if necessary to Deposit Account No. **09-0460** in the name of International Business Machines Corporation. A duplicate copy of this sheet is attached.

Respectfully submitted,

November 30, 2007

Date



Paul D. Greeley
Registration No. 31,019
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.
One Landmark Square, 10th Floor
Stamford, CT 06901-2682
Telephone: (203) 327-4500
Telefax: (203) 327-6401

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellants: Soria et al.

Serial No.: 09/960,769

For: IMPLEMENTING VERSIONING SUPPORT FOR DATA USING A TWO-
TABLE APPROACH THAT MAXIMIZES DATABASE EFFICIENCY

Filed: 21 SEP 2001

Examiner: Cesar B. Paula

Art Unit: 2178

Confirmation No.: 6311

Customer No.: 64612

Attorney Docket Nos.: STL920000113US1
1232.002USU

APPEAL BRIEF FILED UNDER 35 U.S.C. §134

Mail Stop Appeal Brief - Patents
Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellants are filing this Appeal Brief under 35 U.S.C. §134 and in accordance with the provisions of 37 C.F.R. §41.37(a), and believe that the Appeal Brief complies with the requirements set forth in 37 C.F.R. §41.37(c). The claims on appeal are set forth in an Appendix, below.

Appellants do not believe that a petition or fee for an extension of time is required to file this Appeal Brief. However, should Appellants be mistaken, please consider this to be a petition for any required extension of time, and please then also charge Deposit Account No. 09/0460, in the name of International Business Machines Corporation, for the required fee. Likewise, the Commissioner is hereby authorized to charge Deposit Account No. 09/0460 for any required fee not submitted herewith, or submitted incorrectly, so as to maintain the pendency of the above-identified patent application.

(1) Real Party in Interest

The real party in interest is International Business Machines Corporation.

(2) Related Appeals and Interferences

The undersigned attorney is not aware of any related appeals or interferences.

(3) Status of the Claims

Claims 1 – 8 and 10 - 43 are pending in this application, and are the subject of this Appeal. The claims can be found below in an Appendix.

In an Office Action mailed 1 MAY 2007 (hereinafter "the Office Action"), the Examiner made final a rejection of claims 1 – 8 and 10 – 43. In particular:

claims 1 – 8 are rejected;
claim 9 is canceled; and
claims 10 – 43 are rejected.

(4) Status of Amendments

No amendments to any of the claims were proposed subsequent to the mailing of the Office Action.

(5) Summary of Claimed Subject Matter

This Summary makes reference to FIGS. 1 and 2. These figures are provided below, at the end of the Summary.

The present method and system supports versioning of database data by associating a version number (i.e., an identifier) having a value with a database data item, establishing a table for storing a most recent version of the data item, establishing a second table for storing all versions of the data item other than the most recent version, storing the current version of the data item in the first table, storing all other versions of the data item in the second table, and determining the version of the database data item based on the version number and storage location of the database data item.

FIG. 1 is a distributed data processing system environment suitable for supporting implementation the present invention.

FIG. 2 is a depiction of a two-table approach for supporting versioning of data in accord with the present invention, wherein the most recent version of the stored data items are stored in one table and all versions of the data items other than the most recent version are stored in the second table.

In FIG. 1, there is depicted a representation of a distributed data processing system 100. A library database server 45 operates as a centralized data storage repository for data tables used for the storage of data. Data may be stored in a central storage memory 55.

FIG. 2 depicts an exemplary organization of database 200 data for supporting the versioning of data items (datum) using two tables for storing data by library database server 45, in accordance with the present invention. Database 200 data is stored in two tables, namely, a first table 205, (Table 1), and a second table 210, (Table 2). First table 205 is established and maintained for storing the most recent version of a database data item therein. Second table 210 is established and maintained for storing all other versions of the data item other than the most recent version of the data item therein.

An attribute, preferably a version number is associated with a data item in order to support the versioning of the data item, and hence the data in database 200. The user of system environment 100 is presented with the version number and can input the version number into system environment 100. From the user's perspective, the version number (intuitively) starts at 1 and is incremented for subsequent versions. The version number is preferably incremented from one (1) to a maximum value, m.

For the purpose of being able to accurately reference a particular version of the data item in database 200, library database server 45 is enabled to track the versions of data stored in library database server 45 and other locations such as storage devices 20 (FIG. 1). Library database server 45 uses a (version number-1) value to account for the versioned data. The (version number-1) value is generated by system 100 implementing the present invention and associated with the data item. The (version number-1) value starts at zero (0) and is incremented therefrom for subsequent versions of the data item. Accordingly, (version number-1) values can be generated by system 100 by incrementing the value of the maximum (version number-1) value currently maintained by the system for the data item by 1.

As shown in FIG. 2, data located in database 200 is represented as "DATA ITEM XX (YY)", where XX indicates a particular data item entity, and YY indicates the (version number-1) value of the data item entity. For example, the first instance (i.e., version) of data item 1, 215a, stored in database 200 in library database server 45 is internally associated with (version number-1) value zero (0). The next version of data item 1 stored in database 200, 215b, is internally associated with or assigned (version number-1) value one (1). All other subsequent versions of data item 1 (e.g., 215c and 215d) are associated with the (version number-1) value incremented in a similar manner. There are five versions of data item 1 in database 200, namely, four older versions, 215a, 215b, 215c, 215d associated with (version number-1) values 0, 1, 2, 3, respectively, and the most recent version of data item 1, 215, associated with (version number-1) value (4).

The most recent version of data item 1, that is, internal (version number-1) value equals 4 (i.e., DATA ITEM 1 (4)) is stored in first table 205 at 215. All versions of DATA ITEM 1 other than the most recent version are stored in second table 210, see data items 215(a-d).

Regarding DATA ITEM 2, DATA ITEM 3, and DATA ITEM 4, the most recent version of each of these data items is also stored in first table 205 (e.g., 220, 225, 230), whereas all versions of the data items (1-4) other than the most recent version of each (e.g., 220(a-b), 230(a-e)) are stored in second table 210. In this manner, the partitioning and storage of data into two tables ensures that the most recent version is in first table 205 and all other versions of the data, if any, are stored in second table 210.

There is only one instance of exemplary DATA ITEM 3 stored in database 200, see FIG. 2, DATA ITEM 3 (0), reference numeral 225. In accordance with the above discussion, it is noted that this single instance, and hence the only and most recent version of DATA ITEM 3, is stored in first table 205.

Regarding DATA ITEM 4, there are the maximum number ($n=m-1$) versions of DATA ITEM 4 located in database 200. Since the (version number-1) values generated by library database server 45 increment the (version number-1) value associated with data item 4 by 1 starting from zero (0) up to the maximum number n , the most recent version of data item 4 (i.e., DATA ITEM 4 (n), 230) is stored in first table 205 and all versions other than the most recent version of data item 4 (i.e., DATA ITEM 4 (0) through DATA ITEM 4 ($n-1$)) are stored in second table 210.

An important consequence of storing the most recent version of data items 215, 220, 225, and 230 in first table 205 and all other versions of corresponding data in second table 210 is that manipulations and operations performed on database 200 data can be improved and/or optimized due, at least in part, to the partitioning of the versions of data into first and second tables, 205 and 210, respectively. In an operational environment the most frequently accessed data is very often the most recently stored data. As such, an operation, for example a request for the retrieval of data from database 200, is statistically likely to be a request to retrieve the most recently stored version of the data item specified in the request for retrieval. By partitioning the data into first table 205 for storing the most recent version, and second table 210 for storing all other versions of the data item, if any, the efficiency, throughput, and execution of operations and manipulations of data can be improved since

the querying of the two tables can be optimized based on the version (i.e., most recent and/or old version(s)) of the data item to being searched.

A number of operations can be executed on database 200, such as, queries and other data manipulations. The SQL DML operations Insert, Retrieve, etc. may be used in conjunction with the system and method of the present invention.

An example of a SQL operation executed on database 200 data includes an Insert command. The Insert command is used to add records to an existing table in database 200. Execution of a first instance of an Insert command to insert a record into database 200 inserts the specified data into first table 205. For executing subsequent Insert commands on data stored in first table 205, the existing most recent version of the data item in first table 205 is copied to second table 210. The copy of the data still residing in first table 205 is then updated in accord with the Insert command and parameters specified therein. Thus, first table 205 is updated to contain the most recent version of the data as augmented by the Insert command, and second table 210 contains the older version(s) of the data.

The systems and/or methods disclosed herein may be implemented by a computer readable storage media (e.g., CD-ROM 48, FIG. 1) having media having program instructions embodied therein for executing the methods of the present invention, and in turn carried out by a processor.

Concise Explanation Of The Subject Matter Defined In Each Of The Independent Claims

The present application contains three independent claims, namely claims 1, 17 and 31. Below, Appellants are providing a concise explanation of the subject matter defined in each of the independent claims. The explanation refers to the specification by page and line number, and to FIGS. 1 and 2, by reference characters.

Claim 1 provides for a method for supporting versioning of data in a content management system. The method comprises:

maintaining a first table 205 for storing an identifier of a most recent version of a data item (page 5, lines 26 – 27); and

maintaining a second table 210 for storing an identifier of an older version of said data item (page 5, lines 29 – 30),
wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table (page 11, lines 19 – 21), and (ii) said first table is updated to identify a new version of said data item (page 11, lines 21 – 24).

Claim 17 provides for a system 45 for supporting versioning of data in a content management system (page 4, lines 20 – 21 and page 5, lines 23 – 25). The system comprises:

a memory 55 (page 4, lines 17 – 18);
a module 200 that maintains (a) a first table 205 for storing an identifier of a most recent version of a data item in said memory, and (b) a second table 210 for storing an identifier of an older version of said data item in said memory (page 5, lines 23 – 29),
wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table (page 11, lines 19 – 21), and (ii) said first table is updated to identify a new version of said data item (page 11, lines 21 – 24).

Claim 31 provides for a storage medium 48 having computer readable program instructions embodied therein for supporting versioning of data in a content management system (page 18, lines 18 – 22). The storage medium comprises:

program instructions for maintaining a first table 205 for storing an identifier of a most recent version of a data item (page 5, lines 26 – 27);
program instructions for maintaining a second table 210 for storing an identifier of an older version of said data item (page 5, lines 29 – 30); and
program instructions for performing an operation, wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table (page 11, lines 19 – 21), and (ii) said first table is updated to identify a new version of said data item (page 11, lines 21 – 24).

1/3
SORIA et al.
STL920000113US1 CHS

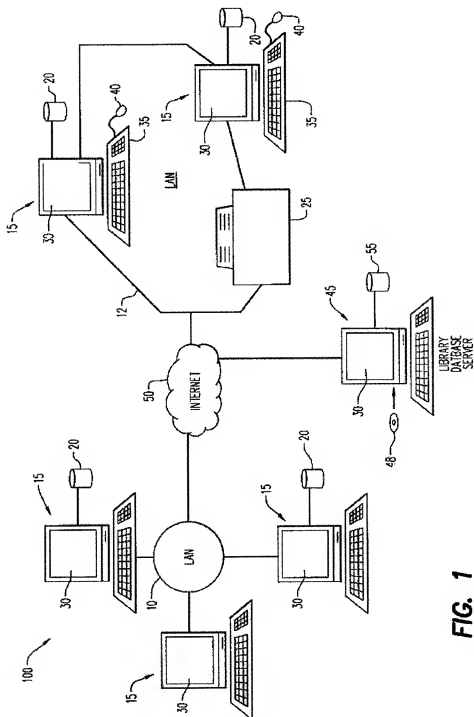


FIG. 1

Replacement Sheet
U.S. Application Serial No. 09/960,769
Soria et al
2/3

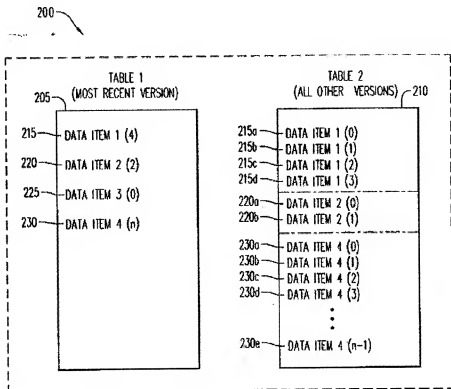


FIG. 2

(6) Grounds of Rejection to be Reviewed on Appeal

The first issue presented for review is the propriety of the Examiner's rejection of claims 1 - 6, 8, 10, 12 - 15, 17 - 22, 24, 25, 27 - 29, 31 - 36, 38, 39, 41 and 42 under 35 U.S.C. 102(b) as being anticipated by International Publication No. WO 99/08206 to Sinander (hereinafter "the Sinander publication").

The second issue presented for review is the propriety of the Examiner's rejection of claims 7, 23 and 37 under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of U.S. Patent No. 6,591,342 to Akkary et al. (hereinafter "the Akkary et al. patent").

The third issue presented for review is the propriety of the Examiner's rejection of claims 11, 26 and 40 under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of U.S. Patent Application Publication No. 20020103815 to Duvillier et al. (hereinafter "the Duvillier et al. publication").

The fourth issue presented for review is the propriety of the Examiner's rejection of claims 16, 30 and 43 under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of U.S. Patent Application Publication No. 20020073089 to Schwartz et al. (hereinafter "the Schwartz et al. publication").

(7) Argument

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

In order to anticipate a claim under Section 102 and render it unpatentable, a single prior art reference must not only expressly or inherently disclose each and every element set forth in the claim, but the reference must also be enabling, i.e. it must clearly put the claimed subject matter in the possession of the public. *In re Brown*, 329 F. 2d 1066, 144 USPQ 245 (CCPA 1964); *In re Kalm*, 378 F. 2d 959, 1154 USPQ 10 (CCPA 1967); *Rem-Cru Titanium, Inc. v. Watson*, 147 F. Supp. 915 112

USPQ 88 (Dist. Ct., DC 1956); *Akzo N.V. v. United States ITC*, 808 F.2d 1471, 1 USPQ 2d 1241 (Fed. Cir. 1986); and *In re Spada*, 911 F.2d 705, 15 USPQ 2d 1655 (Fed. Cir. 1990). The mere disclosure of concepts does not anticipate.

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959).

If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984)

Furthermore, if an independent claim is nonobvious under 35 U.S.C. §103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

(a) Claims 1 – 8 and 10 - 43 stand or fall together.

First Issue: Rejection of claims 1 - 6, 8, 10, 12 - 15, 17 - 22, 24, 25, 27 - 29, 31 - 36, 38, 39, 41 and 42

In section 3 of the Office Action, claims 1 - 6, 8, 10, 12 - 15, 17 - 22, 24, 25, 27 - 29, 31 - 36, 38, 39, 41 and 42 are rejected under 35 U.S.C. 102(b) as being anticipated by the Sinander publication.

As mentioned above, claim 1 provides for a method for supporting versioning of data in a content management system. The method includes maintaining a first table for storing an identifier of a most recent version of a data item, and maintaining a second table for storing an identifier of an older version of the data item. When the data item is to be updated, (i) the second table is updated to include the identifier of the most recent version of the data from the first table, and (ii) the first table is updated to identify a new version of the data item.

The Sinander application is directed toward a method for upgrading a database that uses a table (also referred to as "the old table") for storing data and a stored procedure (also referred to as "the previous version of the stored procedure") for processing the data that is stored in the (old) table (page 2, lines 28 – 30). The data that is stored in the (old) table is copied to a new table (page 2, lines 34 – 35). A new version of the stored procedure is added to the database (page 2, lines 36 – 37). An additional stored procedure is added to the database, where the additional stored procedure causes processing to take place in accordance with both of the previous version of the stored procedure and the new version of the stored procedure (page 3, lines 1 – 7). Thus, during the upgrade of the database, both of the old version of the stored procedure and the new version of the stored procedure are executed in parallel (page 3, line 22).

The Sinander publication, with reference to FIG. 2b, describes a new table created to receive data that is stored in an old table (page 5, lines 9 - 12). The old table and the new table contain data to be updated, and during the upgrade, the old table and the new table are synchronized, that is both of the old table and the new table are updated (to hold updated data) (page 6, lines 7 -15).

The Sinander publication also discloses a systemtable (e.g., Table 1) that holds references to stored procedures (e.g., base version, target version, and upgrade version) (page 7, lines 1 - 22). The base version of a procedure is used during normal operation (page 7, lines 23 - 25). The target version of a procedure is used during an upgrade operation (page 8, lines 1 - 6). The upgrade version facilitates the updating of the old (data) table and the new (data) table, e.g., the tables of FIG. 2b, (page 8, lines 6 - 13).

Although the Sinander publication, in FIG. 2b discloses the old table and the new table, and on page 7 discloses the systemtable, the Sinander publication nonetheless fails to disclose the elements of claim 1.

In the Sinander publication, in FIG. 2b, the old table and the new table each holds data, rather than an identifier of a data item. As such, the old table and the new table of FIG. 2b are not a disclosure of a first table for storing **an identifier** of a most recent version of a data item, and, a second table for storing **an identifier** of an older version of said data item, as recited in claim 1.

With regard to the use of the systemtable, the Sinander publication does not disclose a second systemtable. Nevertheless, Appellants considered that in the Sinander publication, the target version column of Table 1 could be regarded as a first table, and the base version column of Table 1 could be regarded as a second table. However, even under such an interpretation, the Sinander publication does not teach that the base version column is updated from the target version column. Consequently, the Sinander publication does not disclose that said **second table (for storing an identifier of an older version) is updated** to include said identifier of said most recent version of said data **from said first table (for storing an identifier of a most recent version)**, as recited in claim 1.

In view of the foregoing, Appellants submits that the Sinander publication does not anticipate claim 1.

Claims 17 and 31 each include recitals similar to that of claim 1, as described above. Accordingly, claims 17 and 31, for reasoning similar to that provided in support of claim 1, are also novel over the Sinander publication.

Claims 2 – 6, 8, 10 and 12 – 15 depend from claim 1. Claims 18 – 22, 24, 25 and 27 – 29 depend from claim 17. Claims 32 – 36, 38, 39, 41 and 42 depend from claim 31. By virtue of these dependencies, claims 2 – 6, 8, 10, 12 – 15, 18 – 22, 24, 25, 27 – 29, 32 – 36, 38, 39, 41 and 42 are also novel over the Sinander publication.

Appellants respectfully request reconsideration and withdrawal of the section 102(b) rejection of claims 1 – 6, 8, 10, 12 – 15, 17 – 22, 24, 25, 27 – 29, 31 – 36, 38, 39, 41 and 42.

Second Issue: Rejection of claims 7, 23 and 37

In section 4 of the Office Action, claims 7, 23 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of the Akkary et al. patent.

Claims 7, 23 and 37 depend from claims 1, 17 and 31, respectively. Appellants submit that the Akkary et al. patent does not make up for the deficiency of the Sinander publication, as the Sinander publication relates to claims 1, 17 and 31. Accordingly, Appellants further submit that claims 1, 17 and 31, and claims 7, 23 and 37, by virtue of their dependencies, are all patentable over the cited combination of the Sinander publication and the Akkary et al. patent.

Appellants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 7, 23 and 37.

Third Issue: Rejection of claims 11, 26 and 40

In section 5 of the Office Action, claims 11, 26 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of the Duvillier et al. publication.

Claims 11, 26 and 40 depend from claims 1, 17 and 31, respectively. Appellants submit that the Duvillier et al. publication does not make up for the deficiency of the Sinander publication, as the Sinander publication relates to claims 1, 17 and 31. Accordingly, Appellants further submit that claims 1, 17 and 31, and claims 11, 26 and 40, by virtue of their dependencies, are all patentable over the cited combination of the Sinander publication and the Duvillier et al. publication.

Appellants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 11, 26 and 40.

Fourth Issue: Rejection of claims 16, 30 and 43

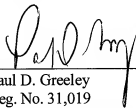
In section 6 of the Office Action, claims 16, 30 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of the Schwartz et al. publication.

Claims 16, 30 and 43 depend from claims 1, 17 and 31, respectively. Appellants submit that the Schwartz et al. publication does not make up for the deficiency of the Sinander publication, as the Sinander publication relates to claims 1, 17 and 31. Accordingly, Appellants further submit that claims 1, 17 and 31, and claims 16, 30 and 43, by virtue of their dependencies, are all patentable over the cited combination of the Sinander publication and the Schwartz et al. publication.

Appellants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 16, 30 and 43.

In view of the foregoing arguments, Appellants respectfully requests that the Board of Appeals reverse the rejection of claims 1 - 8 and 10 - 43.

Respectfully submitted,



Paul D. Greeley
Reg. No. 31,019
Attorney for the Appellant
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.
One Landmark Square, 10th Floor
Stamford, CT 06901-2682
Tel: 203-327-4500
Fax: 203-327-6401

Date

11/30/07

(8) Claims Appendix

The claims on appeal are set forth below.

1. (previously presented) A method for supporting versioning of data in a content management system, said method comprising:
maintaining a first table for storing an identifier of a most recent version of a data item; and
maintaining a second table for storing an identifier of an older version of said data item,
wherein, when said data item is to be updated, (i) said second table is updated to include said
identifier of said most recent version of said data from said first table, and (ii) said first
table is updated to identify a new version of said data item.
2. (previously presented) The method of claim 1, further comprising associating different version numbers with different versions of said data item.
3. (previously presented) The method of claim 2, wherein each of said different versions is associated with a (version number – 1) value.
4. (previously presented) The method of claim 3, wherein a particular version of said data item is determined based on an associated one of said (version number – 1) values.
5. (previously presented) The method of claim 3, further comprising generating said (version number -1) value for successive versions of said data item by incrementing said (version number – 1) value from zero (0) to n.
6. (previously presented) The method of claim 2, further comprising generating a value for successive versions of said data item by incrementing said version number from zero (0) to m.
7. (original) The method of claim 6, wherein m has a predetermined maximum value.

8. (previously presented) The method of claim 1, wherein a version number having a value of zero (0) is associated with said most recent version of said data item or an oldest version of said data item, depending on a context of use for said version number.

9. (canceled)

10. (previously presented) The method of claim 1, wherein an operation including a version number having a value of zero (0) is interpreted as a request for said most recent version of said stored data item, and said operation is selected from a group consisting of a query operation, a retrieve operation, and an update operation.

11. (previously presented) The method of claim 1, wherein an operation including a version number having a value of zero (0) is interpreted as a request for an oldest version of said stored data item, and said operation is a delete operation.

12. (previously presented) The method of claim 1, further comprising performing a query for said identifier of said most recent version or said identifier of said older version.

13. (original) The method of claim 1, wherein a first instance of a version of said data item is stored in said first table.

14. (previously presented) The method of claim 1, further comprising performing a query on said first table and said second table, wherein a column attribute of a column selected by said query is retained in a result of said query.

15. (original) The method of claim 14, wherein said query invokes a union operation.

16. (previously presented) The method of claim 14, wherein said column attribute is obtained from a sequential query language description area (SQLDA) of said query result.

17. (previously presented) A system for supporting versioning of data in a content management system, said system comprising:

a memory;

a module that maintains (a) a first table for storing an identifier of a most recent version of a data item in said memory, and (b) a second table for storing an identifier of an older version of said data item in said memory,

wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table, and (ii) said first table is updated to identify a new version of said data item.

18. (previously presented) The system of claim 17, further comprising a module that associates different version numbers with different versions of said data item.

19. (previously presented) The system of claim 18, wherein each of said different versions is associated with a (version number - 1) value.

20. (previously presented) The system of claim 19, wherein a particular version of said data item is determined based on an associated one of said (version number - 1) values.

21. (previously presented) The system of claim 19, further comprising a module that generates said (version number - 1) value for successive versions of said data item by incrementing said (version number - 1) value from zero (0) to n.

22. (previously presented) The system of claim 18, further comprising a module that generates a value for successive versions of said data item by incrementing said version number from zero (0) to m.

23. (original) The system of claim 22, wherein m has a predetermined maximum value.

24. (previously presented) The system of claim 17, wherein a version number having a value of zero (0) is associated with said most recent version of said data item or an oldest version of said data item, depending on a context of use for said version number.

25. (previously presented) The system of claim 17, wherein an operation including a version number having a value of zero (0) input to said system is interpreted as a request for said most recent version of said stored data item, and said operation is selected from a group consisting of a query operation, a retrieve operation, and an update operation.

26. (previously presented) The system of claim 17, wherein an operation including a version number having a value of zero (0) input to said system is interpreted as a request for an oldest version of said stored data item, and said operation is a delete operation.

27. (previously presented) The system of claim 17, wherein a first instance of a version of said data item is stored in said first table.

28. (previously presented) The system of claim 27, wherein a column attribute of a column selected by a query performed on said first table and said second table is retained in a result of said query.

29. (original) The system of claim 28, wherein said query invokes a union operation.

30. (previously presented) The system of claim 28, wherein said column attribute is obtained from a sequential query language description area (SQLDA) of said query result.

31. (previously presented) A storage medium having computer readable program instructions embodied therein for supporting versioning of data in a content management system, said storage medium comprising:

program instructions for maintaining a first table for storing an identifier of a most recent version of a data item;

program instructions for maintaining a second table for storing an identifier of an older version of said data item ; and

program instructions for performing an operation, wherein, when said data item is to be updated,

(i) said second table is updated to include said identifier of said most recent version of said

data from said first table, and (ii) said first table is updated to identify a new version of said data item.

32. (previously presented) The storage medium of claim 31, further comprising program instructions for associating different version numbers with different versions of said data item.

33. (previously presented) The storage medium of claim 32, comprising program instructions for associating each of said different versions with a (version number - 1) value.

34. (previously presented) The storage medium of claim 33, wherein a particular version of said data item is determined based on an associated one of said (version number - 1) values.

35. (previously presented) The storage medium of claim 33, comprising program instructions for generating said (version number -1) value for successive versions of said data item by incrementing said (version number - 1) value from zero (0) to n.

36. (previously presented) The storage medium of claim 32, comprising program instructions for generating a value for successive versions of said data item by incrementing said version number from zero (0) to m.

37. (original) The storage medium of claim 36, wherein m has a predetermined maximum value.

38. (previously presented) The storage medium of claim 31, comprising program instructions for associating a version number having a value of zero (0) with said most recent version of said stored data item or an oldest version of said stored data item, depending on a context of use for said version number.

39. (previously presented) The storage medium of claim 31, comprising program instructions for interpreting an operation including a version number having a value of zero (0) as a request for said most recent version of said stored data item, wherein said operation is selected from a group consisting of a query operation, a retrieve operation, and an update operation.

40. (previously presented) The storage medium of claim 31, comprising program instructions for interpreting an operation including a version number having a value of zero (0) as a request for an oldest version of said stored data item, and said operation is a delete operation.

41. (original) The storage medium of claim 31, comprising program instructions for retaining a column attribute of a column selected by a query performed on said first table and said second table.

42. (original) The storage medium of claim 41, wherein said query invokes a union operation.

43. (previously presented) The method of claim 41, wherein said column attribute is obtained from a sequential query language description area (SQLDA) of said query result.

(9) Evidence Appendix

None.

(10) Related Proceedings Appendix

None.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellants: Soria et al.
Serial No.: 09/960,769
For: IMPLEMENTING VERSIONING SUPPORT FOR DATA USING A TWO-
TABLE APPROACH THAT MAXIMIZES DATABASE EFFICIENCY
Filed: 21 SEP 2001
Examiner: Cesar B. Paula
Art Unit: 2178
Confirmation No.: 6311
Customer No.: 64612

Attorney Docket Nos.: STL920000113US1
1232.002USU

APPEAL BRIEF FILED UNDER 35 U.S.C. §134

Mail Stop Appeal Brief - Patents
Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellants are filing this Appeal Brief under 35 U.S.C. §134 and in accordance with the provisions of 37 C.F.R. §41.37(a), and believe that the Appeal Brief complies with the requirements set forth in 37 C.F.R. §41.37(c). The claims on appeal are set forth in an Appendix, below.

Appellants do not believe that a petition or fee for an extension of time is required to file this Appeal Brief. However, should Appellants be mistaken, please consider this to be a petition for any required extension of time, and please then also charge Deposit Account No. 09/0460, in the name of International Business Machines Corporation, for the required fee. Likewise, the Commissioner is hereby authorized to charge Deposit Account No. 09/0460 for any required fee not submitted herewith, or submitted incorrectly, so as to maintain the pendency of the above-identified patent application.

(1) Real Party in Interest

The real party in interest is International Business Machines Corporation.

(2) Related Appeals and Interferences

The undersigned attorney is not aware of any related appeals or interferences.

(3) Status of the Claims

Claims 1 – 8 and 10 - 43 are pending in this application, and are the subject of this Appeal. The claims can be found below in an Appendix.

In an Office Action mailed 1 MAY 2007 (hereinafter "the Office Action"), the Examiner made final a rejection of claims 1 – 8 and 10 – 43. In particular:

claims 1 – 8 are rejected;
claim 9 is canceled; and
claims 10 – 43 are rejected.

(4) Status of Amendments

No amendments to any of the claims were proposed subsequent to the mailing of the Office Action.

(5) Summary of Claimed Subject Matter

This Summary makes reference to FIGS. 1 and 2. These figures are provided below, at the end of the Summary.

The present method and system supports versioning of database data by associating a version number (i.e., an identifier) having a value with a database data item, establishing a table for storing a most recent version of the data item, establishing a second table for storing all versions of the data item other than the most recent version, storing the current version of the data item in the first table, storing all other versions of the data item in the second table, and determining the version of the database data item based on the version number and storage location of the database data item.

FIG. 1 is a distributed data processing system environment suitable for supporting implementation the present invention.

FIG. 2 is a depiction of a two-table approach for supporting versioning of data in accord with the present invention, wherein the most recent version of the stored data items are stored in one table and all versions of the data items other than the most recent version are stored in the second table.

In FIG. 1, there is depicted a representation of a distributed data processing system 100. A library database server 45 operates as a centralized data storage repository for data tables used for the storage of data. Data may be stored in a central storage memory 55.

FIG. 2 depicts an exemplary organization of database 200 data for supporting the versioning of data items (datum) using two tables for storing data by library database server 45, in accordance with the present invention. Database 200 data is stored in two tables, namely, a first table 205, (Table 1), and a second table 210, (Table 2). First table 205 is established and maintained for storing the most recent version of a database data item therein. Second table 210 is established and maintained for storing all other versions of the data item other than the most recent version of the data item therein.

An attribute, preferably a version number is associated with a data item in order to support the versioning of the data item, and hence the data in database 200. The user of system environment 100 is presented with the version number and can input the version number into system environment 100. From the user's perspective, the version number (intuitively) starts at 1 and is incremented for subsequent versions. The version number is preferably incremented from one (1) to a maximum value, m.

For the purpose of being able to accurately reference a particular version of the data item in database 200, library database server 45 is enabled to track the versions of data stored in library database server 45 and other locations such as storage devices 20 (FIG. 1). Library database server 45 uses a (version number-1) value to account for the versioned data. The (version number-1) value is generated by system 100 implementing the present invention and associated with the data item. The (version number-1) value starts at zero (0) and is incremented therefrom for subsequent versions of the data item. Accordingly, (version number-1) values can be generated by system 100 by incrementing the value of the maximum (version number-1) value currently maintained by the system for the data item by 1.

As shown in FIG. 2, data located in database 200 is represented as "DATA ITEM XX (YY)", where XX indicates a particular data item entity, and YY indicates the (version number-1) value of the data item entity. For example, the first instance (i.e., version) of data item 1, 215a, stored in database 200 in library database server 45 is internally associated with (version number-1) value zero (0). The next version of data item 1 stored in database 200, 215b, is internally associated with or assigned (version number-1) value one (1). All other subsequent versions of data item 1 (e.g., 215c and 215d) are associated with the (version number-1) value incremented in a similar manner. There are five versions of data item 1 in database 200, namely, four older versions, 215a, 215b, 215c, 215d associated with (version number-1) values 0, 1, 2, 3, respectively, and the most recent version of data item 1, 215, associated with (version number-1) value (4).

The most recent version of data item 1, that is, internal (version number-1) value equals 4 (i.e., DATA ITEM 1 (4)) is stored in first table 205 at 215. All versions of DATA ITEM 1 other than the most recent version are stored in second table 210, see data items 215(a-d).

Regarding DATA ITEM 2, DATA ITEM 3, and DATA ITEM 4, the most recent version of each of these data items is also stored in first table 205 (e.g., 220, 225, 230), whereas all versions of the data items (1-4) other than the most recent version of each (e.g., 220(a-b), 230(a-c)) are stored in second table 210. In this manner, the partitioning and storage of data into two tables ensures that the most recent version is in first table 205 and all other versions of the data, if any, are stored in second table 210.

There is only one instance of exemplary DATA ITEM 3 stored in database 200, see FIG. 2, DATA ITEM 3 (0), reference numeral 225. In accordance with the above discussion, it is noted that this single instance, and hence the only and most recent version of DATA ITEM 3, is stored in first table 205.

Regarding DATA ITEM 4, there are the maximum number ($n=m-1$) versions of DATA ITEM 4 located in database 200. Since the (version number-1) values generated by library database server 45 increment the (version number-1) value associated with data item 4 by 1 starting from zero (0) up to the maximum number n , the most recent version of data item 4 (i.e., DATA ITEM 4 (n), 230) is stored in first table 205 and all versions other than the most recent version of data item 4 (i.e., DATA ITEM 4 (0) through DATA ITEM 4 ($n-1$)) are stored in second table 210.

An important consequence of storing the most recent version of data items 215, 220, 225, and 230 in first table 205 and all other versions of corresponding data in second table 210 is that manipulations and operations performed on database 200 data can be improved and/or optimized due, at least in part, to the partitioning of the versions of data into first and second tables, 205 and 210, respectively. In an operational environment the most frequently accessed data is very often the most recently stored data. As such, an operation, for example a request for the retrieval of data from database 200, is statistically likely to be a request to retrieve the most recently stored version of the data item specified in the request for retrieval. By partitioning the data into first table 205 for storing the most recent version, and second table 210 for storing all other versions of the data item, if any, the efficiency, throughput, and execution of operations and manipulations of data can be improved since

the querying of the two tables can be optimized based on the version (i.e., most recent and/or old version(s)) of the data item to being searched.

A number of operations can be executed on database 200, such as, queries and other data manipulations. The SQL DML operations Insert, Retrieve, etc. may be used in conjunction with the system and method of the present invention.

An example of a SQL operation executed on database 200 data includes an Insert command. The Insert command is used to add records to an existing table in database 200. Execution of a first instance of an Insert command to insert a record into database 200 inserts the specified data into first table 205. For executing subsequent Insert commands on data stored in first table 205, the existing most recent version of the data item in first table 205 is copied to second table 210. The copy of the data still residing in first table 205 is then updated in accord with the Insert command and parameters specified therein. Thus, first table 205 is updated to contain the most recent version of the data as augmented by the Insert command, and second table 210 contains the older version(s) of the data.

The systems and/or methods disclosed herein may be implemented by a computer readable storage media (e.g., CD-ROM 48, FIG. 1) having media having program instructions embodied therein for executing the methods of the present invention, and in turn carried out by a processor.

Concise Explanation Of The Subject Matter Defined In Each Of The Independent Claims

The present application contains three independent claims, namely claims 1, 17 and 31. Below, Appellants are providing a concise explanation of the subject matter defined in each of the independent claims. The explanation refers to the specification by page and line number, and to FIGS. 1 and 2, by reference characters.

Claim 1 provides for a method for supporting versioning of data in a content management system. The method comprises:

maintaining a first table 205 for storing an identifier of a most recent version of a data item (page 5, lines 26 – 27); and

maintaining a second table 210 for storing an identifier of an older version of said data item (page 5, lines 29 – 30),

wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table (page 11, lines 19 – 21), and (ii) said first table is updated to identify a new version of said data item (page 11, lines 21 – 24).

Claim 17 provides for a system 45 for supporting versioning of data in a content management system (page 4, lines 20 – 21 and page 5, lines 23 – 25). The system comprises:

a memory 55 (page 4, lines 17 – 18);

a module 200 that maintains (a) a first table 205 for storing an identifier of a most recent version of a data item in said memory, and (b) a second table 210 for storing an identifier of an older version of said data item in said memory (page 5, lines 23 – 29),

wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table (page 11, lines 19 – 21), and (ii) said first table is updated to identify a new version of said data item (page 11, lines 21 – 24).

Claim 31 provides for a storage medium 48 having computer readable program instructions embodied therein for supporting versioning of data in a content management system (page 18, lines 18 – 22). The storage medium comprises:

program instructions for maintaining a first table 205 for storing an identifier of a most recent version of a data item (page 5, lines 26 – 27);

program instructions for maintaining a second table 210 for storing an identifier of an older version of said data item (page 5, lines 29 – 30); and

program instructions for performing an operation, wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table (page 11, lines 19 – 21), and (ii) said first table is updated to identify a new version of said data item (page 11, lines 21 – 24).

1/3
SORIA et al.
STL920000113US1 CHS

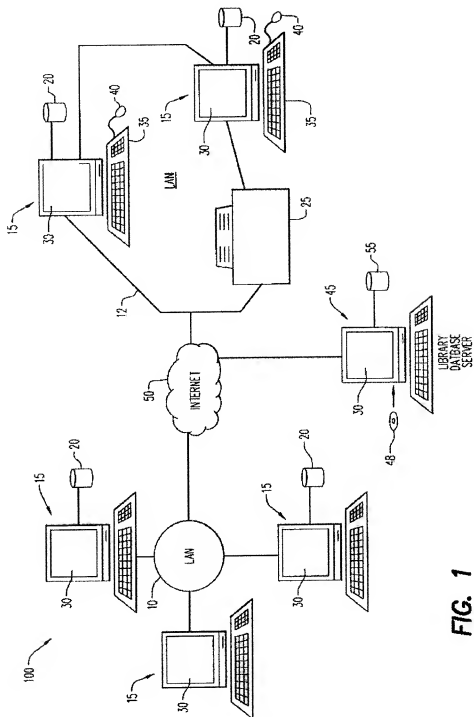


FIG. 1

Replacement Sheet
U.S. Application Serial No. 09/960,769
Sorla et al
2/3

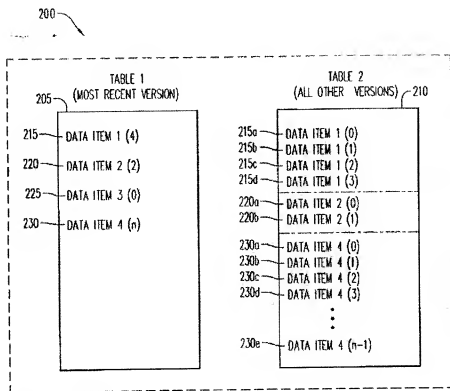


FIG. 2

(6) Grounds of Rejection to be Reviewed on Appeal

The first issue presented for review is the propriety of the Examiner's rejection of claims 1 - 6, 8, 10, 12 - 15, 17 - 22, 24, 25, 27 - 29, 31 - 36, 38, 39, 41 and 42 under 35 U.S.C. 102(b) as being anticipated by International Publication No. WO 99/08206 to Sinander (hereinafter "the Sinander publication").

The second issue presented for review is the propriety of the Examiner's rejection of claims 7, 23 and 37 under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of U.S. Patent No. 6,591,342 to Akkary et al. (hereinafter "the Akkary et al. patent").

The third issue presented for review is the propriety of the Examiner's rejection of claims 11, 26 and 40 under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of U.S. Patent Application Publication No. 20020103815 to Duvillier et al. (hereinafter "the Duvillier et al. publication").

The fourth issue presented for review is the propriety of the Examiner's rejection of claims 16, 30 and 43 under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of U.S. Patent Application Publication No. 20020073089 to Schwartz et al. (hereinafter "the Schwartz et al. publication").

(7) Argument

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

In order to anticipate a claim under Section 102 and render it unpatentable, a single prior art reference must not only expressly or inherently disclose each and every element set forth in the claim, but the reference must also be enabling, i.e. it must clearly put the claimed subject matter in the possession of the public. *In re Brown*, 329 F. 2d 1066, 144 USPQ 245 (CCPA 1964); *In re Kalm*, 378 F. 2d 959, 1154 USPQ 10 (CCPA 1967); *Rem-Cru Titanium, Inc. v. Watson*, 147 F. Supp. 915 112

USPQ 88 (Dist. Ct., DC 1956); *Akzo N.V. v. United States ITC*, 808 F.2d 1471, 1 USPQ 2d 1241 (Fed. Cir. 1986); and *In re Spada*, 911 F.2d 705, 15 USPQ 2d 1655 (Fed. Cir. 1990). The mere disclosure of concepts does not anticipate.

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959).

If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984)

Furthermore, if an independent claim is nonobvious under 35 U.S.C. §103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

(a) Claims 1 – 8 and 10 - 43 stand or fall together.

First Issue: Rejection of claims 1 - 6, 8, 10, 12 - 15, 17 - 22, 24, 25, 27 - 29, 31 - 36, 38, 39, 41 and 42

In section 3 of the Office Action, claims 1 - 6, 8, 10, 12 - 15, 17 - 22, 24, 25, 27 - 29, 31 - 36, 38, 39, 41 and 42 are rejected under 35 U.S.C. 102(b) as being anticipated by the Sinander publication.

As mentioned above, claim 1 provides for a method for supporting versioning of data in a content management system. The method includes maintaining a first table for storing an identifier of a most recent version of a data item, and maintaining a second table for storing an identifier of an older version of the data item. When the data item is to be updated, (i) the second table is updated to include the identifier of the most recent version of the data from the first table, and (ii) the first table is updated to identify a new version of the data item.

The Sinander application is directed toward a method for upgrading a database that uses a table (also referred to as "the old table") for storing data and a stored procedure (also referred to as "the previous version of the stored procedure") for processing the data that is stored in the (old) table (page 2, lines 28 – 30). The data that is stored in the (old) table is copied to a new table (page 2, lines 34 – 35). A new version of the stored procedure is added to the database (page 2, lines 36 – 37). An additional stored procedure is added to the database, where the additional stored procedure causes processing to take place in accordance with both of the previous version of the stored procedure and the new version of the stored procedure (page 3, lines 1 – 7). Thus, during the upgrade of the database, both of the old version of the stored procedure and the new version of the stored procedure are executed in parallel (page 3, line 22).

The Sinander publication, with reference to FIG. 2b, describes a new table created to receive data that is stored in an old table (page 5, lines 9 - 12). The old table and the new table contain data to be updated, and during the upgrade, the old table and the new table are synchronized, that is both of the old table and the new table are updated (to hold updated data) (page 6, lines 7 -15).

The Sinander publication also discloses a systemtable (e.g., Table 1) that holds references to stored procedures (e.g., base version, target version, and upgrade version) (page 7, lines 1 - 22). The base version of a procedure is used during normal operation (page 7, lines 23 - 25). The target version of a procedure is used during an upgrade operation (page 8, lines 1 - 6). The upgrade version facilitates the updating of the old (data) table and the new (data) table, e.g., the tables of FIG. 2b, (page 8, lines 6 - 13).

Although the Sinander publication, in FIG. 2b discloses the old table and the new table, and on page 7 discloses the systemtable, the Sinander publication nonetheless fails to disclose the elements of claim 1.

In the Sinander publication, in FIG. 2b, the old table and the new table each holds data, rather than an identifier of a data item. As such, the old table and the new table of FIG. 2b are not a disclosure of a first table for storing **an identifier** of a most recent version of a data item, and, a second table for storing **an identifier** of an older version of said data item, as recited in claim 1.

With regard to the use of the systemtable, the Sinander publication does not disclose a second systemtable. Nevertheless, Appellants considered that in the Sinander publication, the target version column of Table 1 could be regarded as a first table, and the base version column of Table 1 could be regarded as a second table. However, even under such an interpretation, the Sinander publication does not teach that the base version column is updated from the target version column. Consequently, the Sinander publication does not disclose that said **second table (for storing an identifier of an older version) is updated** to include said identifier of said most recent version of said data **from said first table (for storing an identifier of a most recent version)**, as recited in claim 1.

In view of the foregoing, Appellants submits that the Sinander publication does not anticipate claim 1.

Claims 17 and 31 each include recitals similar to that of claim 1, as described above. Accordingly, claims 17 and 31, for reasoning similar to that provided in support of claim 1, are also novel over the Sinander publication.

Claims 2 – 6, 8, 10 and 12 – 15 depend from claim 1. Claims 18 – 22, 24, 25 and 27 – 29 depend from claim 17. Claims 32 – 36, 38, 39, 41 and 42 depend from claim 31. By virtue of these dependencies, claims 2 – 6, 8, 10, 12 – 15, 18 – 22, 24, 25, 27 – 29, 32 – 36, 38, 39, 41 and 42 are also novel over the Sinander publication.

Appellants respectfully request reconsideration and withdrawal of the section 102(b) rejection of claims 1 – 6, 8, 10, 12 – 15, 17 – 22, 24, 25, 27 – 29, 31 – 36, 38, 39, 41 and 42.

Second Issue: Rejection of claims 7, 23 and 37

In section 4 of the Office Action, claims 7, 23 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of the Akkary et al. patent.

Claims 7, 23 and 37 depend from claims 1, 17 and 31, respectively. Appellants submit that the Akkary et al. patent does not make up for the deficiency of the Sinander publication, as the Sinander publication relates to claims 1, 17 and 31. Accordingly, Appellants further submit that claims 1, 17 and 31, and claims 7, 23 and 37, by virtue of their dependencies, are all patentable over the cited combination of the Sinander publication and the Akkary et al. patent.

Appellants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 7, 23 and 37.

Third Issue: Rejection of claims 11, 26 and 40

In section 5 of the Office Action, claims 11, 26 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of the Duvillier et al. publication.

Claims 11, 26 and 40 depend from claims 1, 17 and 31, respectively. Appellants submit that the Duvillier et al. publication does not make up for the deficiency of the Sinander publication, as the Sinander publication relates to claims 1, 17 and 31. Accordingly, Appellants further submit that claims 1, 17 and 31, and claims 11, 26 and 40, by virtue of their dependencies, are all patentable over the cited combination of the Sinander publication and the Duvillier et al. publication.

Appellants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 11, 26 and 40.

Fourth Issue: Rejection of claims 16, 30 and 43

In section 6 of the Office Action, claims 16, 30 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of the Schwartz et al. publication.

Claims 16, 30 and 43 depend from claims 1, 17 and 31, respectively. Appellants submit that the Schwartz et al. publication does not make up for the deficiency of the Sinander publication, as the Sinander publication relates to claims 1, 17 and 31. Accordingly, Appellants further submit that claims 1, 17 and 31, and claims 16, 30 and 43, by virtue of their dependencies, are all patentable over the cited combination of the Sinander publication and the Schwartz et al. publication.

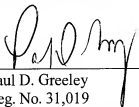
Appellants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 16, 30 and 43.

In view of the foregoing arguments, Appellants respectfully requests that the Board of Appeals reverse the rejection of claims 1 – 8 and 10 - 43.

Respectfully submitted,

Date

11/30/07



Paul D. Greeley
Reg. No. 31,019
Attorney for the Appellant
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.
One Landmark Square, 10th Floor
Stamford, CT 06901-2682
Tel: 203-327-4500
Fax: 203-327-6401

(8) Claims Appendix

The claims on appeal are set forth below.

1. (previously presented) A method for supporting versioning of data in a content management system, said method comprising:
maintaining a first table for storing an identifier of a most recent version of a data item; and
maintaining a second table for storing an identifier of an older version of said data item,
wherein, when said data item is to be updated, (i) said second table is updated to include said
identifier of said most recent version of said data from said first table, and (ii) said first
table is updated to identify a new version of said data item.
2. (previously presented) The method of claim 1, further comprising associating different version numbers with different versions of said data item.
3. (previously presented) The method of claim 2, wherein each of said different versions is associated with a (version number - 1) value.
4. (previously presented) The method of claim 3, wherein a particular version of said data item is determined based on an associated one of said (version number - 1) values.
5. (previously presented) The method of claim 3, further comprising generating said (version number - 1) value for successive versions of said data item by incrementing said (version number - 1) value from zero (0) to n.
6. (previously presented) The method of claim 2, further comprising generating a value for successive versions of said data item by incrementing said version number from zero (0) to m.
7. (original) The method of claim 6, wherein m has a predetermined maximum value.

8. (previously presented) The method of claim 1, wherein a version number having a value of zero (0) is associated with said most recent version of said data item or an oldest version of said data item, depending on a context of use for said version number.

9. (canceled)

10. (previously presented) The method of claim 1, wherein an operation including a version number having a value of zero (0) is interpreted as a request for said most recent version of said stored data item, and said operation is selected from a group consisting of a query operation, a retrieve operation, and an update operation.

11. (previously presented) The method of claim 1, wherein an operation including a version number having a value of zero (0) is interpreted as a request for an oldest version of said stored data item, and said operation is a delete operation.

12. (previously presented) The method of claim 1, further comprising performing a query for said identifier of said most recent version or said identifier of said older version.

13. (original) The method of claim 1, wherein a first instance of a version of said data item is stored in said first table.

14. (previously presented) The method of claim 1, further comprising performing a query on said first table and said second table, wherein a column attribute of a column selected by said query is retained in a result of said query.

15. (original) The method of claim 14, wherein said query invokes a union operation.

16. (previously presented) The method of claim 14, wherein said column attribute is obtained from a sequential query language description area (SQLDA) of said query result.

17. (previously presented) A system for supporting versioning of data in a content management system, said system comprising:

a memory;

a module that maintains (a) a first table for storing an identifier of a most recent version of a data item in said memory, and (b) a second table for storing an identifier of an older version of said data item in said memory,

wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table, and (ii) said first table is updated to identify a new version of said data item.

18. (previously presented) The system of claim 17, further comprising a module that associates different version numbers with different versions of said data item.

19. (previously presented) The system of claim 18, wherein each of said different versions is associated with a (version number - 1) value.

20. (previously presented) The system of claim 19, wherein a particular version of said data item is determined based on an associated one of said (version number - 1) values.

21. (previously presented) The system of claim 19, further comprising a module that generates said (version number - 1) value for successive versions of said data item by incrementing said (version number - 1) value from zero (0) to n.

22. (previously presented) The system of claim 18, further comprising a module that generates a value for successive versions of said data item by incrementing said version number from zero (0) to m.

23. (original) The system of claim 22, wherein m has a predetermined maximum value.

24. (previously presented) The system of claim 17, wherein a version number having a value of zero (0) is associated with said most recent version of said data item or an oldest version of said data item, depending on a context of use for said version number.

25. (previously presented) The system of claim 17, wherein an operation including a version number having a value of zero (0) input to said system is interpreted as a request for said most recent version of said stored data item, and said operation is selected from a group consisting of a query operation, a retrieve operation, and an update operation.

26. (previously presented) The system of claim 17, wherein an operation including a version number having a value of zero (0) input to said system is interpreted as a request for an oldest version of said stored data item, and said operation is a delete operation.

27. (previously presented) The system of claim 17, wherein a first instance of a version of said data item is stored in said first table.

28. (previously presented) The system of claim 27, wherein a column attribute of a column selected by a query performed on said first table and said second table is retained in a result of said query.

29. (original) The system of claim 28, wherein said query invokes a union operation.

30. (previously presented) The system of claim 28, wherein said column attribute is obtained from a sequential query language description area (SQLDA) of said query result.

31. (previously presented) A storage medium having computer readable program instructions embodied therein for supporting versioning of data in a content management system, said storage medium comprising:

program instructions for maintaining a first table for storing an identifier of a most recent version of a data item;

program instructions for maintaining a second table for storing an identifier of an older version of said data item ; and

program instructions for performing an operation, wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said

data from said first table, and (ii) said first table is updated to identify a new version of said data item.

32. (previously presented) The storage medium of claim 31, further comprising program instructions for associating different version numbers with different versions of said data item.

33. (previously presented) The storage medium of claim 32, comprising program instructions for associating each of said different versions with a (version number - 1) value.

34. (previously presented) The storage medium of claim 33, wherein a particular version of said data item is determined based on an associated one of said (version number - 1) values.

35. (previously presented) The storage medium of claim 33, comprising program instructions for generating said (version number - 1) value for successive versions of said data item by incrementing said (version number - 1) value from zero (0) to n.

36. (previously presented) The storage medium of claim 32, comprising program instructions for generating a value for successive versions of said data item by incrementing said version number from zero (0) to m.

37. (original) The storage medium of claim 36, wherein m has a predetermined maximum value.

38. (previously presented) The storage medium of claim 31, comprising program instructions for associating a version number having a value of zero (0) with said most recent version of said stored data item or an oldest version of said stored data item, depending on a context of use for said version number.

39. (previously presented) The storage medium of claim 31, comprising program instructions for interpreting an operation including a version number having a value of zero (0) as a request for said most recent version of said stored data item, wherein said operation is selected from a group consisting of a query operation, a retrieve operation, and an update operation.

40. (previously presented) The storage medium of claim 31, comprising program instructions for interpreting an operation including a version number having a value of zero (0) as a request for an oldest version of said stored data item, and said operation is a delete operation.

41. (original) The storage medium of claim 31, comprising program instructions for retaining a column attribute of a column selected by a query performed on said first table and said second table.

42. (original) The storage medium of claim 41, wherein said query invokes a union operation.

43. (previously presented) The method of claim 41, wherein said column attribute is obtained from a sequential query language description area (SQLDA) of said query result.

(9) Evidence Appendix

None.

(10) Related Proceedings Appendix

None.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellants: Soria et al.
Serial No.: 09/960,769
For: IMPLEMENTING VERSIONING SUPPORT FOR DATA USING A TWO-
TABLE APPROACH THAT MAXIMIZES DATABASE EFFICIENCY
Filed: 21 SEP 2001
Examiner: Cesar B. Paula
Art Unit: 2178
Confirmation No.: 6311
Customer No.: 64612

Attorney Docket Nos.: STL920000113US1
1232.002USU

APPEAL BRIEF FILED UNDER 35 U.S.C. §134

Mail Stop Appeal Brief - Patents
Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellants are filing this Appeal Brief under 35 U.S.C. §134 and in accordance with the provisions of 37 C.F.R. §41.37(a), and believe that the Appeal Brief complies with the requirements set forth in 37 C.F.R. §41.37(c). The claims on appeal are set forth in an Appendix, below.

Appellants do not believe that a petition or fee for an extension of time is required to file this Appeal Brief. However, should Appellants be mistaken, please consider this to be a petition for any required extension of time, and please then also charge Deposit Account No. 09/0460, in the name of International Business Machines Corporation, for the required fee. Likewise, the Commissioner is hereby authorized to charge Deposit Account No. 09/0460 for any required fee not submitted herewith, or submitted incorrectly, so as to maintain the pendency of the above-identified patent application.

(1) Real Party in Interest

The real party in interest is International Business Machines Corporation.

(2) Related Appeals and Interferences

The undersigned attorney is not aware of any related appeals or interferences.

(3) Status of the Claims

Claims 1 – 8 and 10 - 43 are pending in this application, and are the subject of this Appeal. The claims can be found below in an Appendix.

In an Office Action mailed 1 MAY 2007 (hereinafter "the Office Action"), the Examiner made final a rejection of claims 1 – 8 and 10 – 43. In particular:

claims 1 – 8 are rejected;
claim 9 is canceled; and
claims 10 – 43 are rejected.

(4) Status of Amendments

No amendments to any of the claims were proposed subsequent to the mailing of the Office Action.

(5) Summary of Claimed Subject Matter

This Summary makes reference to FIGS. 1 and 2. These figures are provided below, at the end of the Summary.

The present method and system supports versioning of database data by associating a version number (i.e., an identifier) having a value with a database data item, establishing a table for storing a most recent version of the data item, establishing a second table for storing all versions of the data item other than the most recent version, storing the current version of the data item in the first table, storing all other versions of the data item in the second table, and determining the version of the database data item based on the version number and storage location of the database data item.

FIG. 1 is a distributed data processing system environment suitable for supporting implementation the present invention.

FIG. 2 is a depiction of a two-table approach for supporting versioning of data in accord with the present invention, wherein the most recent version of the stored data items are stored in one table and all versions of the data items other than the most recent version are stored in the second table.

In FIG. 1, there is depicted a representation of a distributed data processing system 100. A library database server 45 operates as a centralized data storage repository for data tables used for the storage of data. Data may be stored in a central storage memory 55.

FIG. 2 depicts an exemplary organization of database 200 data for supporting the versioning of data items (datum) using two tables for storing data by library database server 45, in accordance with the present invention. Database 200 data is stored in two tables, namely, a first table 205, (Table 1), and a second table 210, (Table 2). First table 205 is established and maintained for storing the most recent version of a database data item therein. Second table 210 is established and maintained for storing all other versions of the data item other than the most recent version of the data item therein.

An attribute, preferably a version number is associated with a data item in order to support the versioning of the data item, and hence the data in database 200. The user of system environment 100 is presented with the version number and can input the version number into system environment 100. From the user's perspective, the version number (intuitively) starts at 1 and is incremented for subsequent versions. The version number is preferably incremented from one (1) to a maximum value, m.

For the purpose of being able to accurately reference a particular version of the data item in database 200, library database server 45 is enabled to track the versions of data stored in library database server 45 and other locations such as storage devices 20 (FIG. 1). Library database server 45 uses a (version number-1) value to account for the versioned data. The (version number-1) value is generated by system 100 implementing the present invention and associated with the data item. The (version number-1) value starts at zero (0) and is incremented therefrom for subsequent versions of the data item. Accordingly, (version number-1) values can be generated by system 100 by incrementing the value of the maximum (version number-1) value currently maintained by the system for the data item by 1.

As shown in FIG. 2, data located in database 200 is represented as "DATA ITEM XX (YY)", where XX indicates a particular data item entity, and YY indicates the (version number-1) value of the data item entity. For example, the first instance (i.e., version) of data item 1, 215a, stored in database 200 in library database server 45 is internally associated with (version number-1) value zero (0). The next version of data item 1 stored in database 200, 215b, is internally associated with or assigned (version number-1) value one (1). All other subsequent versions of data item 1 (e.g., 215c and 215d) are associated with the (version number-1) value incremented in a similar manner. There are five versions of data item 1 in database 200, namely, four older versions, 215a, 215b, 215c, 215d associated with (version number-1) values 0, 1, 2, 3, respectively, and the most recent version of data item 1, 215, associated with (version number-1) value (4).

The most recent version of data item 1, that is, internal (version number-1) value equals 4 (i.e., DATA ITEM 1 (4)) is stored in first table 205 at 215. All versions of DATA ITEM 1 other than the most recent version are stored in second table 210, see data items 215(a-d).

Regarding DATA ITEM 2, DATA ITEM 3, and DATA ITEM 4, the most recent version of each of these data items is also stored in first table 205 (e.g., 220, 225, 230), whereas all versions of the data items (1-4) other than the most recent version of each (e.g., 220(a-b), 230(a-e)) are stored in second table 210. In this manner, the partitioning and storage of data into two tables ensures that the most recent version is in first table 205 and all other versions of the data, if any, are stored in second table 210.

There is only one instance of exemplary DATA ITEM 3 stored in database 200, see FIG. 2, DATA ITEM 3 (0), reference numeral 225. In accordance with the above discussion, it is noted that this single instance, and hence the only and most recent version of DATA ITEM 3, is stored in first table 205.

Regarding DATA ITEM 4, there are the maximum number ($n=m-1$) versions of DATA ITEM 4 located in database 200. Since the (version number-1) values generated by library database server 45 increment the (version number-1) value associated with data item 4 by 1 starting from zero (0) up to the maximum number n , the most recent version of data item 4 (i.e., DATA ITEM 4 (n), 230) is stored in first table 205 and all versions other than the most recent version of data item 4 (i.e., DATA ITEM 4 (0) through DATA ITEM 4 ($n-1$)) are stored in second table 210.

An important consequence of storing the most recent version of data items 215, 220, 225, and 230 in first table 205 and all other versions of corresponding data in second table 210 is that manipulations and operations performed on database 200 data can be improved and/or optimized due, at least in part, to the partitioning of the versions of data into first and second tables, 205 and 210, respectively. In an operational environment the most frequently accessed data is very often the most recently stored data. As such, an operation, for example a request for the retrieval of data from database 200, is statistically likely to be a request to retrieve the most recently stored version of the data item specified in the request for retrieval. By partitioning the data into first table 205 for storing the most recent version, and second table 210 for storing all other versions of the data item, if any, the efficiency, throughput, and execution of operations and manipulations of data can be improved since

the querying of the two tables can be optimized based on the version (i.e., most recent and/or old version(s)) of the data item to being searched.

A number of operations can be executed on database 200, such as, queries and other data manipulations. The SQL DML operations Insert, Retrieve, etc. may be used in conjunction with the system and method of the present invention.

An example of a SQL operation executed on database 200 data includes an Insert command. The Insert command is used to add records to an existing table in database 200. Execution of a first instance of an Insert command to insert a record into database 200 inserts the specified data into first table 205. For executing subsequent Insert commands on data stored in first table 205, the existing most recent version of the data item in first table 205 is copied to second table 210. The copy of the data still residing in first table 205 is then updated in accord with the Insert command and parameters specified therein. Thus, first table 205 is updated to contain the most recent version of the data as augmented by the Insert command, and second table 210 contains the older version(s) of the data.

The systems and/or methods disclosed herein may be implemented by a computer readable storage media (e.g., CD-ROM 48, FIG. 1) having media having program instructions embodied therein for executing the methods of the present invention, and in turn carried out by a processor.

Concise Explanation Of The Subject Matter Defined In Each Of The Independent Claims

The present application contains three independent claims, namely claims 1, 17 and 31. Below, Appellants are providing a concise explanation of the subject matter defined in each of the independent claims. The explanation refers to the specification by page and line number, and to FIGS. 1 and 2, by reference characters.

Claim 1 provides for a method for supporting versioning of data in a content management system. The method comprises:

maintaining a first table 205 for storing an identifier of a most recent version of a data item (page 5, lines 26 – 27); and

maintaining a second table 210 for storing an identifier of an older version of said data item (page 5, lines 29 – 30),
wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table (page 11, lines 19 – 21), and (ii) said first table is updated to identify a new version of said data item (page 11, lines 21 – 24).

Claim 17 provides for a system 45 for supporting versioning of data in a content management system (page 4, lines 20 – 21 and page 5, lines 23 – 25). The system comprises:

a memory 55 (page 4, lines 17 – 18);

a module 200 that maintains (a) a first table 205 for storing an identifier of a most recent version of a data item in said memory, and (b) a second table 210 for storing an identifier of an older version of said data item in said memory (page 5, lines 23 – 29),

wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table (page 11, lines 19 – 21), and (ii) said first table is updated to identify a new version of said data item (page 11, lines 21 – 24).

Claim 31 provides for a storage medium 48 having computer readable program instructions embodied therein for supporting versioning of data in a content management system (page 18, lines 18 – 22). The storage medium comprises:

program instructions for maintaining a first table 205 for storing an identifier of a most recent version of a data item (page 5, lines 26 – 27);

program instructions for maintaining a second table 210 for storing an identifier of an older version of said data item (page 5, lines 29 – 30); and

program instructions for performing an operation, wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table (page 11, lines 19 – 21), and (ii) said first table is updated to identify a new version of said data item (page 11, lines 21 – 24).

1/3
SORIA et al.
STL920000113US1 CHS

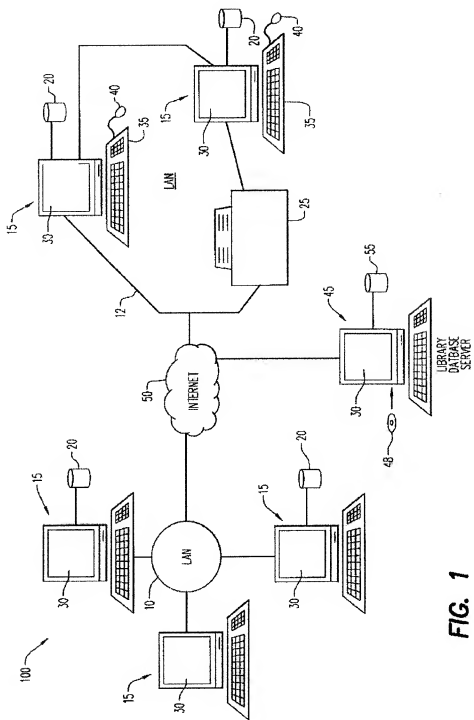


FIG. 1

Replacement Sheet
U.S. Application Serial No. 09/960,769
Sorla et al
2/3

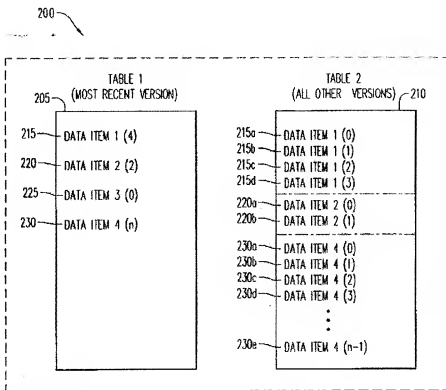


FIG. 2

(6) Grounds of Rejection to be Reviewed on Appeal

The first issue presented for review is the propriety of the Examiner's rejection of claims 1 - 6, 8, 10, 12 - 15, 17 - 22, 24, 25, 27 - 29, 31 - 36, 38, 39, 41 and 42 under 35 U.S.C. 102(b) as being anticipated by International Publication No. WO 99/08206 to Sinander (hereinafter "the Sinander publication").

The second issue presented for review is the propriety of the Examiner's rejection of claims 7, 23 and 37 under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of U.S. Patent No. 6,591,342 to Akkary et al. (hereinafter "the Akkary et al. patent").

The third issue presented for review is the propriety of the Examiner's rejection of claims 11, 26 and 40 under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of U.S. Patent Application Publication No. 20020103815 to Duvillier et al. (hereinafter "the Duvillier et al. publication").

The fourth issue presented for review is the propriety of the Examiner's rejection of claims 16, 30 and 43 under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of U.S. Patent Application Publication No. 20020073089 to Schwartz et al. (hereinafter "the Schwartz et al. publication").

(7) Argument

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

In order to anticipate a claim under Section 102 and render it unpatentable, a single prior art reference must not only expressly or inherently disclose each and every element set forth in the claim, but the reference must also be enabling, i.e. it must clearly put the claimed subject matter in the possession of the public. *In re Brown*, 329 F. 2d 1066, 144 USPQ 245 (CCPA 1964); *In re Kalm*, 378 F. 2d 959, 1154 USPQ 10 (CCPA 1967); *Rem-Cru Titanium, Inc. v. Watson*, 147 F. Supp. 915 112

USPQ 88 (Dist. Ct., DC 1956); *Akzo N.V. v. United States ITC*, 808 F.2d 1471, 1 USPQ 2d 1241 (Fed. Cir. 1986); and *In re Spada*, 911 F.2d 705, 15 USPQ 2d 1655 (Fed. Cir. 1990). The mere disclosure of concepts does not anticipate.

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959).

If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984)

Furthermore, if an independent claim is nonobvious under 35 U.S.C. §103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

(a) Claims 1 - 8 and 10 - 43 stand or fall together.

First Issue: Rejection of claims 1 - 6, 8, 10, 12 - 15, 17 - 22, 24, 25, 27 - 29, 31 - 36, 38, 39, 41 and 42

In section 3 of the Office Action, claims 1 - 6, 8, 10, 12 - 15, 17 - 22, 24, 25, 27 - 29, 31 - 36, 38, 39, 41 and 42 are rejected under 35 U.S.C. 102(b) as being anticipated by the Sinander publication.

As mentioned above, claim 1 provides for a method for supporting versioning of data in a content management system. The method includes maintaining a first table for storing an identifier of a most recent version of a data item, and maintaining a second table for storing an identifier of an older version of the data item. When the data item is to be updated, (i) the second table is updated to include the identifier of the most recent version of the data from the first table, and (ii) the first table is updated to identify a new version of the data item.

The Sinander application is directed toward a method for upgrading a database that uses a table (also referred to as "the old table") for storing data and a stored procedure (also referred to as "the previous version of the stored procedure") for processing the data that is stored in the (old) table (page 2, lines 28 – 30). The data that is stored in the (old) table is copied to a new table (page 2, lines 34 – 35). A new version of the stored procedure is added to the database (page 2, lines 36 – 37). An additional stored procedure is added to the database, where the additional stored procedure causes processing to take place in accordance with both of the previous version of the stored procedure and the new version of the stored procedure (page 3, lines 1 – 7). Thus, during the upgrade of the database, both of the old version of the stored procedure and the new version of the stored procedure are executed in parallel (page 3, line 22).

The Sinander publication, with reference to FIG. 2b, describes a new table created to receive data that is stored in an old table (page 5, lines 9 - 12). The old table and the new table contain data to be updated, and during the upgrade, the old table and the new table are synchronized, that is both of the old table and the new table are updated (to hold updated data) (page 6, lines 7 -15).

The Sinander publication also discloses a system table (e.g., Table 1) that holds references to stored procedures (e.g., base version, target version, and upgrade version) (page 7, lines 1 - 22). The base version of a procedure is used during normal operation (page 7, lines 23 - 25). The target version of a procedure is used during an upgrade operation (page 8, lines 1 - 6). The upgrade version facilitates the updating of the old (data) table and the new (data) table, e.g., the tables of FIG. 2b, (page 8, lines 6 - 13).

Although the Sinander publication, in FIG. 2b discloses the old table and the new table, and on page 7 discloses the systemtable, the Sinander publication nonetheless fails to disclose the elements of claim 1.

In the Sinander publication, in FIG. 2b, the old table and the new table each holds data, rather than an identifier of a data item. As such, the old table and the new table of FIG. 2b are not a disclosure of a first table for storing **an identifier** of a most recent version of a data item, and, a second table for storing **an identifier** of an older version of said data item, as recited in claim 1.

With regard to the use of the systemtable, the Sinander publication does not disclose a second systemtable. Nevertheless, Appellants considered that in the Sinander publication, the target version column of Table 1 could be regarded as a first table, and the base version column of Table 1 could be regarded as a second table. However, even under such an interpretation, the Sinander publication does not teach that the base version column is updated from the target version column. Consequently, the Sinander publication does not disclose that said **second table (for storing an identifier of an older version) is updated** to include said identifier of said most recent version of said data **from said first table (for storing an identifier of a most recent version)**, as recited in claim 1.

In view of the foregoing, Appellants submits that the Sinander publication does not anticipate claim 1.

Claims 17 and 31 each include recitals similar to that of claim 1, as described above. Accordingly, claims 17 and 31, for reasoning similar to that provided in support of claim 1, are also novel over the Sinander publication.

Claims 2 – 6, 8, 10 and 12 – 15 depend from claim 1. Claims 18 – 22, 24, 25 and 27 – 29 depend from claim 17. Claims 32 – 36, 38, 39, 41 and 42 depend from claim 31. By virtue of these dependencies, claims 2 – 6, 8, 10, 12 – 15, 18 – 22, 24, 25, 27 – 29, 32 – 36, 38, 39, 41 and 42 are also novel over the Sinander publication.

Appellants respectfully request reconsideration and withdrawal of the section 102(b) rejection of claims 1 – 6, 8, 10, 12 – 15, 17 – 22, 24, 25, 27 – 29, 31 – 36, 38, 39, 41 and 42.

Second Issue: Rejection of claims 7, 23 and 37

In section 4 of the Office Action, claims 7, 23 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of the Akkary et al. patent.

Claims 7, 23 and 37 depend from claims 1, 17 and 31, respectively. Appellants submit that the Akkary et al. patent does not make up for the deficiency of the Sinander publication, as the Sinander publication relates to claims 1, 17 and 31. Accordingly, Appellants further submit that claims 1, 17 and 31, and claims 7, 23 and 37, by virtue of their dependencies, are all patentable over the cited combination of the Sinander publication and the Akkary et al. patent.

Appellants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 7, 23 and 37.

Third Issue: Rejection of claims 11, 26 and 40

In section 5 of the Office Action, claims 11, 26 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of the Duvillier et al. publication.

Claims 11, 26 and 40 depend from claims 1, 17 and 31, respectively. Appellants submit that the Duvillier et al. publication does not make up for the deficiency of the Sinander publication, as the Sinander publication relates to claims 1, 17 and 31. Accordingly, Appellants further submit that claims 1, 17 and 31, and claims 11, 26 and 40, by virtue of their dependencies, are all patentable over the cited combination of the Sinander publication and the Duvillier et al. publication.

Appellants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 11, 26 and 40.

Fourth Issue: Rejection of claims 16, 30 and 43

In section 6 of the Office Action, claims 16, 30 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Sinander publication in view of the Schwartz et al. publication.

Claims 16, 30 and 43 depend from claims 1, 17 and 31, respectively. Appellants submit that the Schwartz et al. publication does not make up for the deficiency of the Sinander publication, as the Sinander publication relates to claims 1, 17 and 31. Accordingly, Appellants further submit that claims 1, 17 and 31, and claims 16, 30 and 43, by virtue of their dependencies, are all patentable over the cited combination of the Sinander publication and the Schwartz et al. publication.

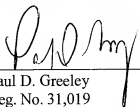
Appellants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 16, 30 and 43.

In view of the foregoing arguments, Appellants respectfully requests that the Board of Appeals reverse the rejection of claims 1 – 8 and 10 - 43.

Respectfully submitted,

Date

11/30/07


Paul D. Greeley
Reg. No. 31,019
Attorney for the Appellant
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.
One Landmark Square, 10th Floor
Stamford, CT 06901-2682
Tel: 203-327-4500
Fax: 203-327-6401

(8) Claims Appendix

The claims on appeal are set forth below.

1. (previously presented) A method for supporting versioning of data in a content management system, said method comprising:
maintaining a first table for storing an identifier of a most recent version of a data item; and
maintaining a second table for storing an identifier of an older version of said data item,
wherein, when said data item is to be updated, (i) said second table is updated to include said
identifier of said most recent version of said data from said first table, and (ii) said first
table is updated to identify a new version of said data item.
2. (previously presented) The method of claim 1, further comprising associating different version numbers with different versions of said data item.
3. (previously presented) The method of claim 2, wherein each of said different versions is associated with a (version number - 1) value.
4. (previously presented) The method of claim 3, wherein a particular version of said data item is determined based on an associated one of said (version number - 1) values.
5. (previously presented) The method of claim 3, further comprising generating said (version number -1) value for successive versions of said data item by incrementing said (version number - 1) value from zero (0) to n.
6. (previously presented) The method of claim 2, further comprising generating a value for successive versions of said data item by incrementing said version number from zero (0) to m.
7. (original) The method of claim 6, wherein m has a predetermined maximum value.

8. (previously presented) The method of claim 1, wherein a version number having a value of zero (0) is associated with said most recent version of said data item or an oldest version of said data item, depending on a context of use for said version number.

9. (canceled)

10. (previously presented) The method of claim 1, wherein an operation including a version number having a value of zero (0) is interpreted as a request for said most recent version of said stored data item, and said operation is selected from a group consisting of a query operation, a retrieve operation, and an update operation.

11. (previously presented) The method of claim 1, wherein an operation including a version number having a value of zero (0) is interpreted as a request for an oldest version of said stored data item, and said operation is a delete operation.

12. (previously presented) The method of claim 1, further comprising performing a query for said identifier of said most recent version or said identifier of said older version.

13. (original) The method of claim 1, wherein a first instance of a version of said data item is stored in said first table.

14. (previously presented) The method of claim 1, further comprising performing a query on said first table and said second table, wherein a column attribute of a column selected by said query is retained in a result of said query.

15. (original) The method of claim 14, wherein said query invokes a union operation.

16. (previously presented) The method of claim 14, wherein said column attribute is obtained from a sequential query language description area (SQLDA) of said query result.

17. (previously presented) A system for supporting versioning of data in a content management system, said system comprising:

a memory;

a module that maintains (a) a first table for storing an identifier of a most recent version of a data item in said memory, and (b) a second table for storing an identifier of an older version of said data item in said memory,

wherein, when said data item is to be updated, (i) said second table is updated to include said identifier of said most recent version of said data from said first table, and (ii) said first table is updated to identify a new version of said data item.

18. (previously presented) The system of claim 17, further comprising a module that associates different version numbers with different versions of said data item.

19. (previously presented) The system of claim 18, wherein each of said different versions is associated with a (version number - 1) value.

20. (previously presented) The system of claim 19, wherein a particular version of said data item is determined based on an associated one of said (version number - 1) values.

21. (previously presented) The system of claim 19, further comprising a module that generates said (version number - 1) value for successive versions of said data item by incrementing said (version number - 1) value from zero (0) to n.

22. (previously presented) The system of claim 18, further comprising a module that generates a value for successive versions of said data item by incrementing said version number from zero (0) to m.

23. (original) The system of claim 22, wherein m has a predetermined maximum value.

24. (previously presented) The system of claim 17, wherein a version number having a value of zero (0) is associated with said most recent version of said data item or an oldest version of said data item, depending on a context of use for said version number.

25. (previously presented) The system of claim 17, wherein an operation including a version number having a value of zero (0) input to said system is interpreted as a request for said most recent version of said stored data item, and said operation is selected from a group consisting of a query operation, a retrieve operation, and an update operation.

26. (previously presented) The system of claim 17, wherein an operation including a version number having a value of zero (0) input to said system is interpreted as a request for an oldest version of said stored data item, and said operation is a delete operation.

27. (previously presented) The system of claim 17, wherein a first instance of a version of said data item is stored in said first table.

28. (previously presented) The system of claim 27, wherein a column attribute of a column selected by a query performed on said first table and said second table is retained in a result of said query.

29. (original) The system of claim 28, wherein said query invokes a union operation.

30. (previously presented) The system of claim 28, wherein said column attribute is obtained from a sequential query language description area (SQLDA) of said query result.

31. (previously presented) A storage medium having computer readable program instructions embodied therein for supporting versioning of data in a content management system, said storage medium comprising:

program instructions for maintaining a first table for storing an identifier of a most recent version of a data item;

program instructions for maintaining a second table for storing an identifier of an older version of said data item ; and

program instructions for performing an operation, wherein, when said data item is to be updated,

(i) said second table is updated to include said identifier of said most recent version of said

data from said first table, and (ii) said first table is updated to identify a new version of said data item.

32. (previously presented) The storage medium of claim 31, further comprising program instructions for associating different version numbers with different versions of said data item.

33. (previously presented) The storage medium of claim 32, comprising program instructions for associating each of said different versions with a (version number - 1) value.

34. (previously presented) The storage medium of claim 33, wherein a particular version of said data item is determined based on an associated one of said (version number - 1) values.

35. (previously presented) The storage medium of claim 33, comprising program instructions for generating said (version number -1) value for successive versions of said data item by incrementing said (version number - 1) value from zero (0) to n.

36. (previously presented) The storage medium of claim 32, comprising program instructions for generating a value for successive versions of said data item by incrementing said version number from zero (0) to m.

37. (original) The storage medium of claim 36, wherein m has a predetermined maximum value.

38. (previously presented) The storage medium of claim 31, comprising program instructions for associating a version number having a value of zero (0) with said most recent version of said stored data item or an oldest version of said stored data item, depending on a context of use for said version number.

39. (previously presented) The storage medium of claim 31, comprising program instructions for interpreting an operation including a version number having a value of zero (0) as a request for said most recent version of said stored data item, wherein said operation is selected from a group consisting of a query operation, a retrieve operation, and an update operation.

40. (previously presented) The storage medium of claim 31, comprising program instructions for interpreting an operation including a version number having a value of zero (0) as a request for an oldest version of said stored data item, and said operation is a delete operation.

41. (original) The storage medium of claim 31, comprising program instructions for retaining a column attribute of a column selected by a query performed on said first table and said second table.

42. (original) The storage medium of claim 41, wherein said query invokes a union operation.

43. (previously presented) The method of claim 41, wherein said column attribute is obtained from a sequential query language description area (SQLDA) of said query result.

(9) Evidence Appendix

None.

(10) Related Proceedings Appendix

None.